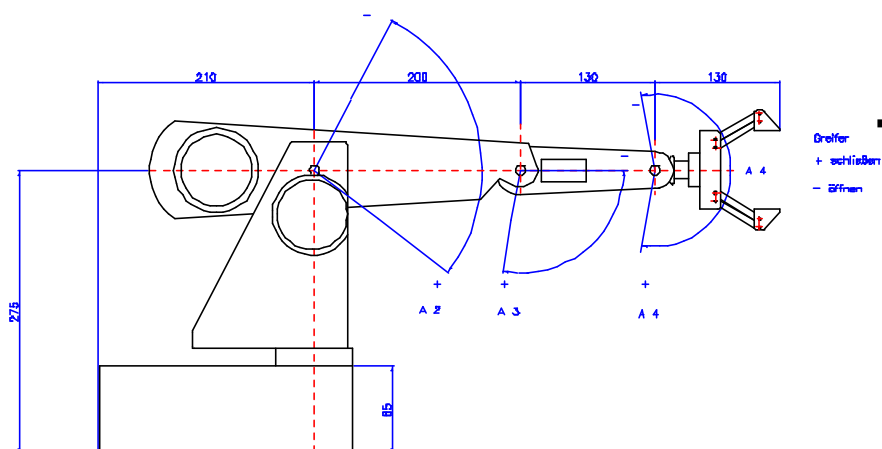
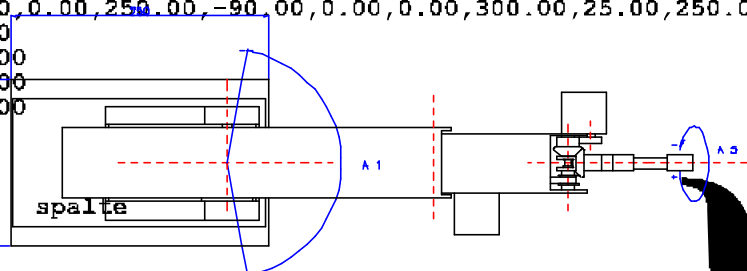


TR 5

ROBOT MANUAL

Version 2.4e

```
PD,pall,275.00,0.00,250.00,-90.00,0.00,0.00,300.00,25.00,250.00,-90.00,0.00,0.00
LET,zeile,1.00
LET,spalte,1.00
LET,zeimax,4.00
LET,spamax,4.00
REF
GR,0
LB,start
PC,pall,zeile
C,entnahme
C,ablage
ADD,zeile,zeile 1.00
IFL,
```



1 Copyright

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, (photocopying) recording or otherwise, without the prior written permission of *P@P*.

Care has been taken in the preparation of this manual. However, nothing contained in it modifies in any way the terms and conditions of the agreement under which this software package including all documentation was acquired, nor increases in any way *P@P* liability. In no event shall *P@P* be liable for incidental or consequential damage in connection with or resulting from the use of the product, this document, or any related material.

The authorised user of this manual shall acquire no rights in respect of trade marks or other intellectual property rights referred to herein which are and shall remain the property of the owner thereof.

P@P Picking Systems GmbH

Schafhofstraße 10
D-90411 Zirndorf

Telephone: (++49)-911-955 116-0

Fax: (++49)-911-955 116-11

eMail: info@pp-systeme.com

Homepage: www.pp-systeme.com

2 Contents	
1	COPYRIGHT2
2	CONTENTS3
3	INTRODUCTION6
3.1	THE MECHANICS6
3.2	THE CONTROLLER.....7
3.3	THE GRIPPER7
3.4	THE TEACHBOX.....8
3.5	TBPS.....8
3.6	PSI.....9
3.7	POWER SUPPLY UNIT9
3.8	INTERFACE CABLE10
3.9	I/O TEST BOX10
3.10	I/O CARD FOR PC.....11
4	TECHNICAL SPECIFICATIONS12
5	DIMENSIONS AND AXIS NUMBERS.....16
6	WORKING RANGE OF THE ROBOT.....17
7	THE TR 5 TEACHBOX.....18
7.1	NUMERICAL KEYS18
7.2	COMMAND KEYS18
7.3	INSTRUCTION KEYS18
8	OPERATING MODES19
8.1	INPUT MODE19
8.2	POSITION MODE.....19
8.3	RUN MODE19
8.4	BREAK MODE.....19
8.5	STEP MODE19
8.6	DISPLAY MODE19
9	SUMMARY OF ALL INSTRUCTIONS AND COMMANDS20
9.1	INSTRUCTIONS20
9.2	COMMANDS20
10	REFERENCE TO THE INDIVIDUAL INSTRUCTIONS21
10.1	PROGRAM START21
10.2	AXIS POSITIONING21
10.3	LOOPS22
10.4	I/O INSTRUCTIONS22
10.5	SET DELAY22
10.6	PROGRAM HALT23
10.7	PROGRAM END23
10.8	DISPLAY MODE23
10.9	INSERT INSTRUCTION.....23
10.10	DELETE INSTRUCTION.....23
10.11	APPENDING INSTRUCTIONS24
10.12	OVERWRITING INSTRUCTIONS24
10.13	PROGRAM RUN / BREAK.....24

11	HARDWARE INSTALLATION, REPACKAGING	25
11.1	HARDWARE REQUIREMENTS.....	25
11.2	INSTALLATION	25
11.3	PACKING THE ROBOT.....	25
12	DIGITAL INPUTS AND OUTPUTS, EMERGENCY STOP FUNCTION	26
12.1	DIGITAL INPUTS (TTL)	26
12.2	OUTPUTS (TTL).....	26
12.3	EMERGENCY STOP FOR THE TR 5.....	26
12.4	POWER SUPPLY.....	27
13	EXERCISES WITH THE TEACHBOX.....	28
13.1	PROGRAM HEADER AND PROGRAM END	29
14	PRACTICE PROGRAMS	30
14.1.1	<i>PROGRAM 1 Single Axis Positioning.....</i>	<i>30</i>
14.1.2	<i>PROGRAM 2: Consecutive Single Axis Positioning for all 6 Axes</i>	<i>31</i>
14.1.3	<i>PROGRAM 3: Simultaneous Multiple Axis Positioning.....</i>	<i>33</i>
14.1.4	<i>PROGRAM 4: Sample Program.....</i>	<i>34</i>
14.2	TIPS AND SUGGESTIONS	35
15	INCLUDE FILES	36
15.1	SUPPORTED COMPILERS AND INTERPRETERS.....	36
15.1.1	<i>Summary of the TR 5 Include Files</i>	<i>36</i>
15.1.1.1	BASIC-Files	36
15.1.1.2	PASCAL-Files.....	37
15.1.1.3	C-Files.....	37
15.1.2	<i>How to use the header files.....</i>	<i>37</i>
15.2	THE BASIC INCLUDE SUBPROGRAMS	38
15.2.1	<i>SUBPROGRAM 1: Set Specified Robot Axis Position</i>	<i>38</i>
15.2.2	<i>SUBPROGRAM 2: Set all Robot Axis Positions</i>	<i>39</i>
15.2.3	<i>SUBPROGRAM 3: Read Specified Robot Axis Position.....</i>	<i>39</i>
15.2.4	<i>SUBPROGRAM 4: Read all Robot Axis Positions</i>	<i>39</i>
15.2.5	<i>SUBPROGRAM 5: Read Input Port Value.....</i>	<i>39</i>
15.2.6	<i>SUBPROGRAM 6: Write Byte to Output Port.....</i>	<i>39</i>
15.2.7	<i>SUBPROGRAM 7: Output Port Byte AND Parameter Byte.....</i>	<i>40</i>
15.2.8	<i>SUBPROGRAM 8: Output Port Byte OR Parameter Byte.....</i>	<i>40</i>
15.2.9	<i>SUBPROGRAM 9: Output Port Byte XOR Parameter Byte.....</i>	<i>40</i>
15.2.10	<i>SUBPROGRAM 10: Set Output Port Bit.....</i>	<i>40</i>
15.2.11	<i>SUBPROGRAM 11: Clear Output Port Bit.....</i>	<i>40</i>
15.2.12	<i>SUBPROGRAM 12: Initialise PC Interface.....</i>	<i>41</i>
15.2.13	<i>SUBPROGRAM 13: Initialise TR 5 Interface.....</i>	<i>41</i>
15.2.14	<i>SUBPROGRAMS 14 and 15:</i>	<i>41</i>
15.3	THE PASCAL AND C INCLUDE PROCEDURES	41
15.3.1	<i>PROCEDURE 1: Set Specified Robot Axis Position</i>	<i>41</i>
15.3.2	<i>PROCEDURE 2: Set all Robot Axis Positions.....</i>	<i>42</i>
15.3.3	<i>PROCEDURE 3: Read Specified Robot Axis Position</i>	<i>42</i>
15.3.4	<i>PROCEDURE 4: Read all Robot Axis Positions</i>	<i>42</i>
15.3.5	<i>PROCEDURE 5: Read Input Port Value.....</i>	<i>42</i>
15.3.6	<i>PROCEDURE 6: Write Byte to Output Port</i>	<i>43</i>
15.3.7	<i>PROCEDURE 7: Output Port Byte AND Parameter Byte.....</i>	<i>43</i>
15.3.8	<i>PROCEDURE 8: Output Port Byte OR Parameter Byte.....</i>	<i>43</i>
15.3.9	<i>PROCEDURE 9: Output Port Byte XOR Parameter Byte.....</i>	<i>43</i>
15.3.10	<i>PROCEDURE 10: Set Output Port Bit.....</i>	<i>43</i>
15.3.11	<i>PROCEDURE 11: Clear Output Port Bit.....</i>	<i>44</i>
15.3.12	<i>PROCEDURE 12: Initialise PC Interface.....</i>	<i>44</i>
15.3.13	<i>PROCEDURE 13: Initialise TR 5 Interface.....</i>	<i>44</i>

16	THE VACUUM GRIPPER	45
16.1	THE VACUUM-GRIPPER CONSISTS OF THE FOLLOWING COMPONENTS:	45
16.2	MOUNTING AND CONNECTING THE ITEMS	45
16.3	RUNNING THE SYSTEM	45
16.4	THE CONTROL BOX.....	46
16.5	PROGRAMMING THE VACUUM GRIPPER WITH TEACHBOX AND TBPS	46
16.6	PROGRAMMING THE VACUUM GRIPPER WITH <i>PSI</i>	47
17	I/O-CARD SMARTLAB	48
17.1	GENERAL INFORMATION	48
17.2	STANDARD SCOPE OF DELIVERY:	48
17.3	TECHNICAL DATA:	49
17.4	COMMISSIONING.....	49
17.5	INSTALLATION	50
17.6	CONFIGURING <i>PSI</i>	51
17.7	CONFIGURING TBPS FOR WINDOWS.....	52
17.8	DRAWINGS.....	52
18	LIMITED WARRANTY TERMS AND CONDITIONS	55
18.1	SHIPPING ADDRESS FOR SERVICE AND REPAIR.....	55
19	INDEX OF FIGURES.....	56
20	INDEX OF TABLES	57

3 Introduction

This manual is valid for the *TR 5* as well as for the *ROB 3* robots of *EUROBTEC*. In order to keep it simple only the expression *TR 5* will be used.

The *TR 5* is a high-quality training robot which has demonstrated its reliability for several years in all spheres of education and training. While undergoing a continuous process of improvements to the model's reliability, this low-cost buckling arm robot has lost nothing of its attractiveness.

The *TR 5*'s articulated arm has 5 rotary axes and a parallel gripper giving several application possibilities. As the controller is integrated in the base of the robot, an easy and fast installation and implementation can be assured.

With the easy-to-use Teachbox and the corresponding software *TBPS*, the beginner quickly becomes familiar with this system and obtains remarkable results. But that's not all. For the already more advanced user we offer *PSI*, a programming system that has been successfully tried and tested in industry. *PSI* really leaves no wishes open.

3.1 The mechanics

The mechanics of the *TR 5* consists of aluminium parts screwed together. Each of the five independent axes is driven by a DC Servo-motor. At axes no. 1,2, and 5 power transmission is assured by toothed-gears. For power transmission at axes 3 and 4, one toothed belt each is used in addition. The absolute positions of all axes are determined by potentiometric position transducers. This out-standing feature of the *TR 5* guarantees that even after the switch-on of the unit the robot controller knows the exact current position of each single axis. Thus, tiresome reference runs are no longer necessary.

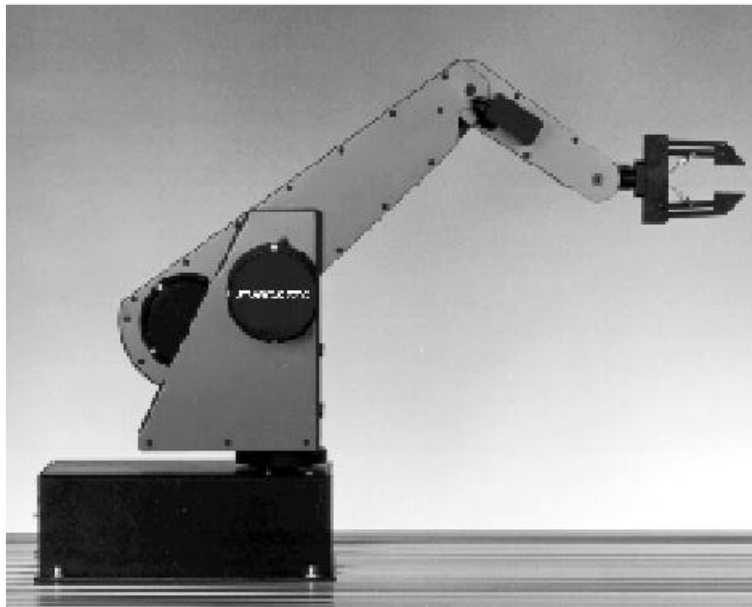


Figure 3-1 TR 5

3.2 The controller

The controller is integrated in the base of the robot. An 8031 processor is the heart of the controller. In the PtP (point-to-point) mode, movements of single axes or even of more axes at the same time can be realised. User programmes can be stored in a battery-backed-up RAM. Memory capacity is 8 kB, this corresponds to approximately 1000 program steps. In addition, the robot can be connected with a PC or ATARI computer via an RS 232 interface. 8 digital, TTL-compatible inputs and outputs are at the user's disposal through a parallel port connector.

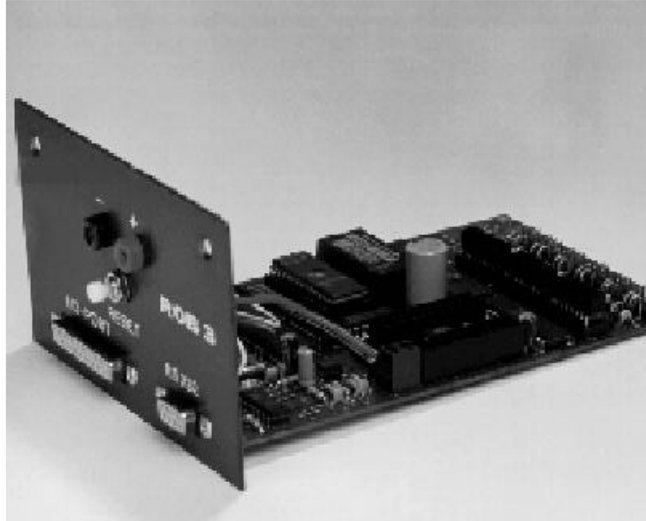


Figure 3-2 The TR 5 controller

3.3 The gripper

The gripper is powered by a DC Servo-motor via a spring-mounted cable. Thanks to this structure, the gripper position as well as the gripping force are freely-programmable.

In addition, the extremely wide stroke of 60 mm permits the handling of a wide variety of pieces without making modifications at the gripper fingers necessary.

For users who wish to handle very thin workpieces having a smooth surface as for example sheets of paper, the standard gripper can be equipped with a vacuum gripper.

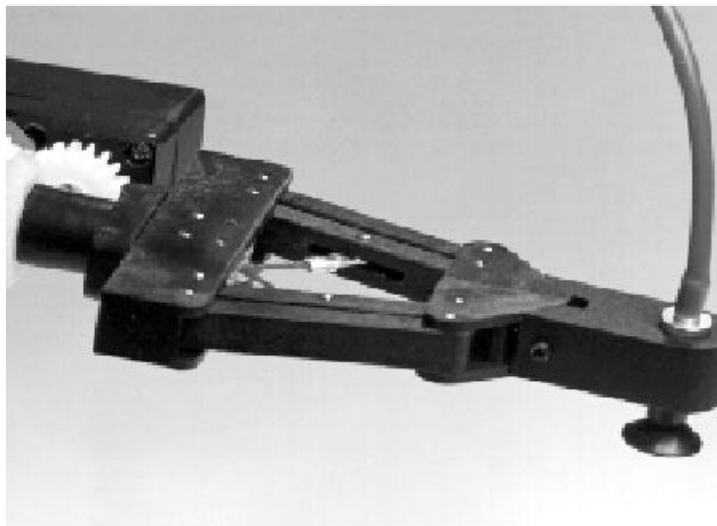


Figure 3-3 The standard gripper equipped with the vacuum option

3.4 The Teachbox

Using the Teachbox the robot can be programmed without the need for a PC. This teach pendant is connected to the I/O-port of the controller. Thanks to its easy-to-learn instruction set, it is ideally suited for the first steps in the field of robotics.



Figure 3-4 The hand held Teachbox

3.5 TBPS

The Teachbox-Programming System is a menu-guided programming software that basically contains the functions of the Teachbox.

An attractive feature of *TBPS* is its ability to later edit programs written via the Teachbox on the screen and to load them afterwards back to the memory of the robot.

TBPS is available for IBM-compatible PCs and ATARI computers. For users who wish to develop their own robot control programs, software interfaces are available in the form of Include-files for the programming languages BASIC, TURBO-PASCAL and C for IBM-compatible PCs.

3.6 **PSI**

The Programming System for Industrial robots is an extremely user-friendly programming language for the ambitious user. The following characteristics represent only an extract of the extensive instruction set:

- Axis-, world- and tool co-ordinate system
- PtP, linear- and circular interpolation
- Convenient subroutines
- Programming variables
- Palletising functions
- Various program test functions

PSI may be run Off-line with all features operational. This low-priced alternative allows robot programming at several work stations for later testing On-line with a robot.

The context-related On-line help-system permits the user to fetch at any time all necessary information from the memory.

PSI runs on IBM-compatible PCs.

3.7 **Power supply unit**



Figure 3-5 The power supply for TR 5

3.8 Interface cable

A 9- or 25-pin serial interface cable enables the TR 5 to be connected to a PC.



Figure 3-6 The RS 232 interface cables

3.9 I/O Test Box

The Test Box permits the simulation and the testing of the 8 digital In-and Outputs.

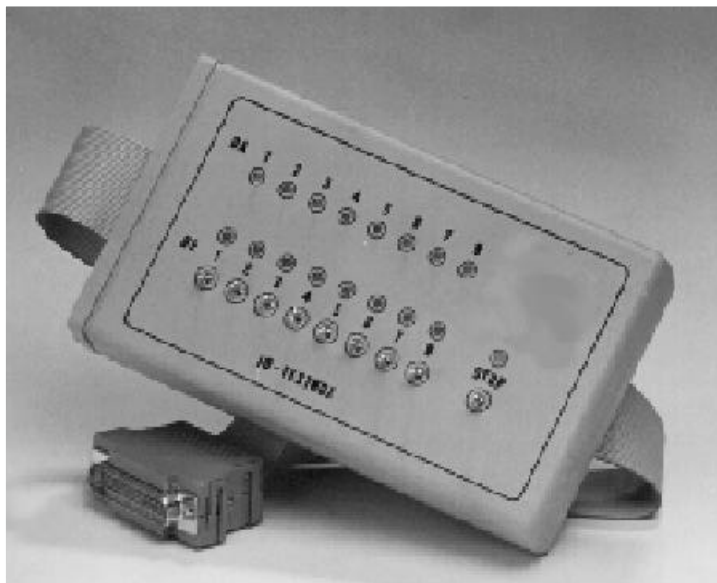


Figure 3-7 The I/O Testbox

3.10 I/O Card for PC

with 16 opto-isolated digital inputs and 16 potential-free relay outputs (max. 24V/1A).

The I/O card is featured only for use in connection with **TBPS** vor Windows, *PSI* or the Software library.

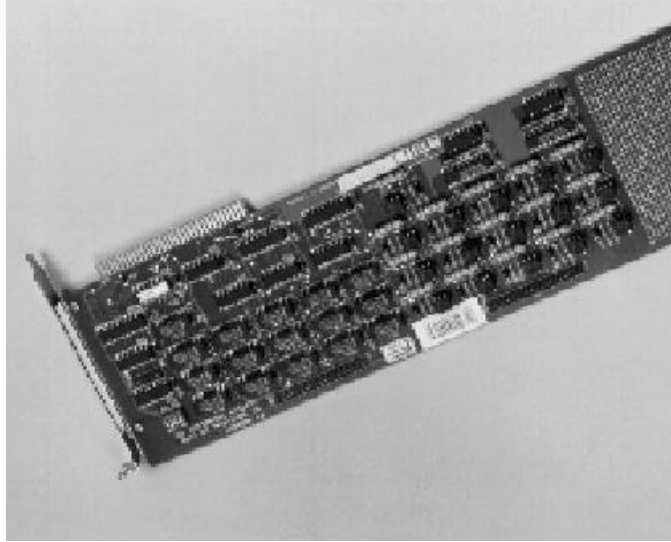


Figure 3-8 The PC I/O-Card

For more details please contact your dealer or P@P.

4 Technical Specifications

MECHANICS	
Kinematics	5-axis link arm robot
Drive systems	DC Servo-motors
Position detection	absolute, potentiometric
Working ranges:	
Axis 1 (base)	160°
Axis 2 (shoulder)	100°
Axis 3 (elbow)	100°
Axis 4 (wrist)	200°
Axis 5 (tool)	200°
Gripper	60 mm
Grippers	electric parallel gripper stroke 0-60 mm, position and gripping force freely programmable (standard version) ----- ---- vacuum gripper with vacuum pump (option)
Max. payload	250 g
Repeatability	+/- 1 mm
Resolution:	
Axis 1	256 increments
Axis 2	256 increments
Axis 3	256 increments
Axis 4	256 increments
Axis 5	256 increments
Gripper	256 increments
Max. joint speeds:	
Axis 1	46 degrees/sec.
Axis 2	40 degrees/sec.
Axis 3	100 degrees/sec.
Axis 4	174 degrees/sec.
Axis 5	176 degrees/sec.
Total weight:	approx. 6 kg (including controller)

Table 4-1

CONTROLLER	
Processor	8031
Axes	5 plus gripper
Controller type	point-to-point (PtP)
Program memory	RAM 8 kB, battery-backed-up
Programmable by	Teachbox, ----- ---- TBPS Software, ----- ---- PSI Software, ----- ---- Software Library (Basic, Turbo Pascal, Microsoft C)
Interfaces: serial (communication with PC) parallel	RS 232 compatible ----- ---- 8 digital inputs, 8 digital outputs (TTL-compatible, active LOW)
Alternative to the parallel interface (for use in connection with <i>PSI</i> or the software libraries only !): digital inputs digital outputs	16 potential-free, 24V/20 mA 16 potential-free 24V/1A via separate plug-in PC-card
Controller dimensions W*D*H	approx. 120mm*170mm*80mm (the controller is integrated in the robot base)

Table 4-2

POWER SUPPLY	
Control system and Axes	9V/3A DC
Mains supply	230-240V/0,1kVA AC

Table 4-3

<i>TR 5 used with Software TBPS</i>	
Programming language	<i>TBPS (Teachbox Programming System)</i>
Hardware configuration	IBM compatible PC starting from XT types, 10 MHz
Operating system	MS-DOS version 3.2 and later
Programming types	On-line ----- Off-line (with only small restrictions)
Co-ordinate systems	axis co-ordinates ----- world co-ordinates
Motion types	asynchron-PtP ----- linear (multipoint)
Teaching possibilities	in axis co-ordinates
Speed ranges	selectable in 5 steps
Compatibility	<i>TBPS</i> -programs are Teachbox compatible. Programs created with the Teachbox can be processed by <i>TBPS</i> and vice versa
Further software components included with delivery	Include-Files for the following programming languages: Q-Basic Turbo-Pascal Microsoft-C
Available on request	interface protocol for communication controller-PC

Table 4-4

<i>TR 5 used with Software PSI</i>	
Programming language	<i>PSI (Programming System for Industrial robots)</i>
Hardware configuration	IBM compatible PC starting from 386 SX /25 MHz types (co-processor recommended)
Operating system	MS-DOS version 3.2 and later
Programming types	On-line ----- Off-line
Co-ordinate systems	axis co-ordinates ----- world co-ordinates ----- tool co-ordinates
Motion types	PtP (point-to-point) ----- linear (multipoint) ----- circular (multipoint)
Overlap	no
Teaching possibilities	in all co-ordinate systems as well as in tool orientation
Structured programming	yes
Variable programming	yes
Integrated palletising function	yes
Help system	context sensitive
Safety functions	1 working region ----- 4 no-go regions
Interrupt processing	with small restrictions

Table 4-5

All axes can be moved simultaneously. The resolution of the axes makes it possible to access a total of 256^5 points within the robot's reach. Gripper opening and closing also has a resolution of 256 steps. The gripper is closed after travel of approximately 200 steps; the remaining steps can be used for setting gripping force.

In contrast to other robots, which always have a fixed home position, the *TR 5* allows you to redefine the start position at the beginning of each program. This makes it possible to use the robot in the most difficult situations imaginable, for example when the home position is obstructed by an immovable object.

5 Dimensions and Axis Numbers

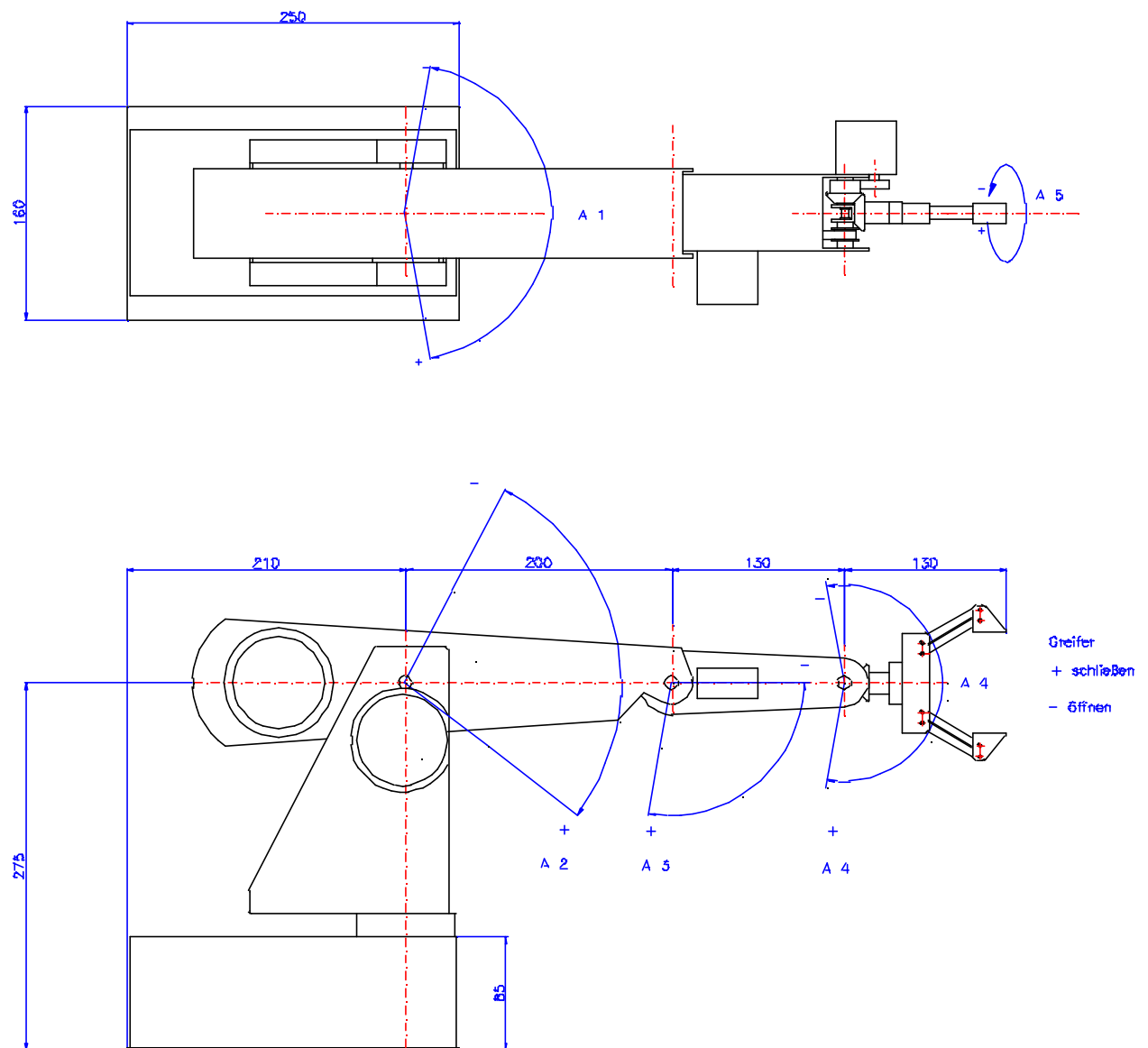


Figure 5-1 Dimensions of TR 5

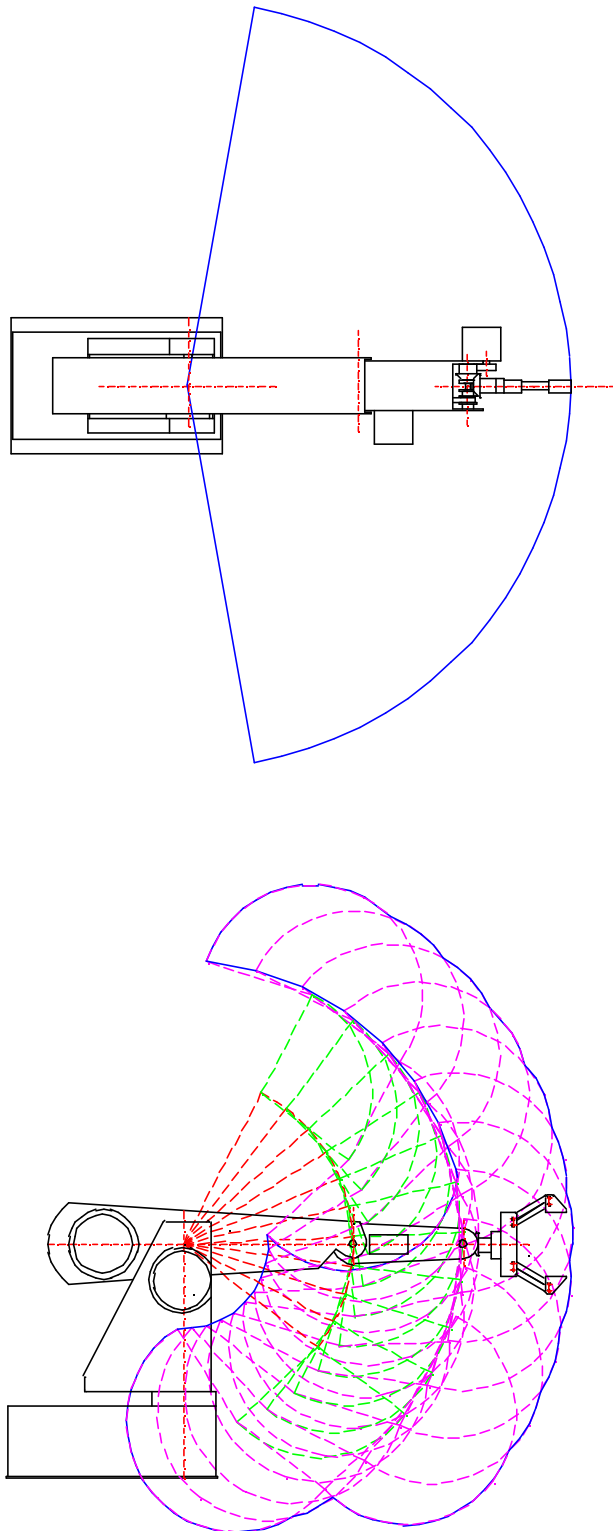
6 Working range of the robot

Figure 6-1 Working range of TR 5

7 The TR 5 Teachbox

The TR 5 Teachbox makes it possible to operate and program the TR 5 without using a computer. It consists of a compact keypad unit with 25 keys, divided into three groups: Numeric keys for entering single or multiple-digit decimal values, command keys for entering directly-executable commands, and instruction keys for entering control program instructions. To program the robot, you simply connect the Teachbox to the TR 5's I/O port; the unit can then be easily disconnected once the program has been started. The female connector on the Teachbox provides all the inputs and outputs necessary for interfacing with the TR 5's I/O port. When the TR 5 is being operated with the Teachbox connected, only 3 digital outputs (DO 1-3) and 5 digital inputs (DI 1-5) are available. Each instruction key is fitted with an LED which lights up to confirm correct entry. If you make an invalid command or instruction entry, the ERR lamp will light up until you clear the incorrect entry by pressing CLR. All instructions and commands must be confirmed with the ENT key.

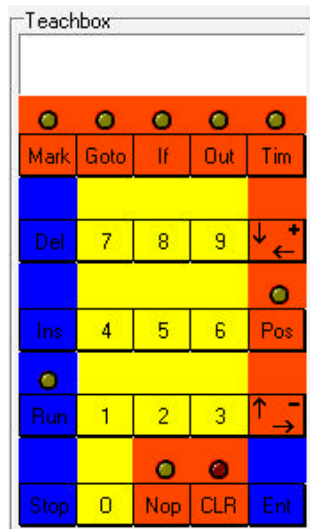


Figure 7-1 Teachbox

7.1 Numerical Keys

Used for entering the axis designation numbers and position values.

7.2 Command Keys

(RUN, STOP, INS, DEL, POS, OUT) : Commands are executed immediately.

7.3 Instruction Keys

(POS, TIM, OUT, MARK, IF, GOTO, NOP) : When you enter instructions, they are stored in the robot's memory in the sequence in which you enter them. They are only executed in RUN or STEP mode.

8 Operating Modes

The robot's integrated control system has six different operating modes.

8.1 INPUT Mode

For creating control programs by entering a sequence of instructions. This is the default mode which is activated whenever the system is powered up.

8.2 POSITION Mode

This mode is selected by pressing one of the numerical keys 1-6, which also selects the corresponding axis. You can then move the axis forwards or backwards with the + and - keys. In this mode, you can also position the axes directly with the sequence POS (axis a) NOP (position n) ENT. Pressing CLR returns you to the INPUT mode.

8.3 RUN Mode

When you enter RUN (label m) ENT, TR 5 starts executing the sequence of instructions in its memory starting at label m.

8.4 BREAK Mode

When the robot is in RUN mode, all keys except the STOP key are disabled. Pressing STOP CLR activates the BREAK mode. You can switch back to RUN mode with RUN ENT (=CONTINUE). Pressing STOP ENT allows you to select any of the other modes.

8.5 STEP Mode

Entering RUN . (label m) ENT switches on the STEP mode, starting at label m (provided that label m exists, of course). You can then run through the program one step at a time by pressing the + key. Pressing CLR exits the STEP mode.

8.6 DISPLAY Mode

Entering STOP NOP (label m) ENT switches the system to the DISPLAY mode, starting at label m. When you press the + key, the next instruction stored in the sequence in memory is displayed (the corresponding LED on the Teachbox lights up), but the instruction is not executed on the TR 5. Pressing CLR exits the DISPLAY mode.

9 Summary of all Instructions and Commands

9.1 Instructions

INSTRUCTION	REMARKS	VALUE RANGE
MARK m ENT	Set label m	m=0..118
POS ENT	Store current position	
TIM t ENT	Set delay period t (x 100 ms)	t=1..65535
GOTO m ENT	Go to label m	endless loop)
GOTO m NOP n ENT	Go to label m, n times	n=0..255
IF i ENT	Wait until input i=low	
IF i NOP m ENT	If input i=low go to label m else continue with next instruction	
OUT k +/- ENT	Set/clear digital output k	k=1..8
NOP ENT	NOP instruction	
STOP 0 ENT	Program start	
INS NOP ENT	Program end	
DEL NOP ENT	Halt instruction, used to separate two consecutive programs in memory	

Table 9-1

9.2 Commands

COMMAND	REMARKS	VALUE RANGE
a +/- ENT	Move axis a in POSITION mode	
POS a NOP n ENT	Move axis a to position n	a=1..6
OUT NOP k +/- ENT	Set/clear digital output k directly	
RUN m ENT	Run from label m	
RUN NOP m ENT	Activate STEP mode starting at label m	
RUN ENT	Continue (BREAK mode)	
RUN NOP ENT	Continue (STEP mode)	
STOP CLR	Break, stop program execution, continue with RUN ENT	
STOP ENT<D>	Stop, switch to INPUT mode	
STOP NOP ENT	Activate DISPLAY mode	
STOP NOP m ENT	Activate DISPLAY mode starting at label m	
INS ENT	Insert instruction	
DEL ENT	Delete instruction	
CLR	Change mode, clear on error	

Table 9-2

10 Reference to the Individual Instructions

10.1 Program Start

STOP 0 ENT

This sequence generates the program header, clearing the contents of the TR 5's memory and initialising it for programming. It should only be entered once, at the beginning of a new programming session.

DON'T use this command when you want to edit programs already in memory or add new programs to the those in memory, as it will completely erase all the instructions and programs already stored in the memory!

MARK 0 ENT

Set label 0, marks beginning of the program.

CLR

Switch to input mode.

This is followed by the body of the program, consisting of positioning sequences, loops and input and output instructions.

10.2 Axis Positioning

(Axis a) +/-

Using this command, you can move axis a up/right or left/down with the +/- keys until it is in the desired position. Since the control program can position several axes at the same time, you can position any number of axes using this technique, until the robot arm is in the desired position.

CLR

Switch to INPUT mode.

POS ENT

Store arm position in memory
or:

POS (axis a) NOP (position n) ENT

Moves axis a directly to position n. Again, you can position any number of axes in this way, until the arm is in the desired position.

CLR

Switch to INPUT mode

POS ENT

Store arm position in memory

10.3 Loops

MARK m ENT

Set label m as loop starting point.

Create the body of the loop by teaching in the arm positioning sequences (see above).

GOTO m ENT

Loop to the starting point label (for endless loops). This option is useful for writing endlessly repeating programs. You can terminate these programs with the STOP function.

GOTO m NOP n ENT

Loop to starting point label n times, thus repeating the instructions in the body of the loop n times. For i=1 to n loop repetitions.

10.4 I/O Instructions

IF i ENT

Stop execution at this point in the program until input i is low.

IF i NOP m ENT

If input i is low when the statement is executed, a jump is performed to label m; otherwise, execution continues with the next instruction.

OUT k +/- ENT

Set/clear output k

Please note the negative TTL logic! (Set=low, clear=high).

OUT NOP k +/- CLR

Allows you to poll the I/O port and the TR 5 periphery directly, either during a programming session or during a break in program execution. Please don't terminate this command with ENT, as this will store it in memory as part of the control program.

10.5 Set Delay

TIM t ENT

Allows you to interrupt program execution for a specified period (t x 100 ms). The range of t is from 0 to 65535, which means that you can program delays of up to 100 minutes. Insert the instruction in a loop to program longer delays.

10.6 Program Halt

DEL NOP ENT

Generates a program halt, and switches on the RUN lamp. Execution can be restarted by pressing ENT. This instruction is also used as a separator between consecutive, separate programs in memory. Pressing CLR switches to INPUT mode.

10.7 Program End

INS NOP ENT

This sequence must always be entered at the end of every teach-in program. When you are editing existing programs, it should never be entered if you make changes in the body of the program only, but you **MUST** enter it if you lengthen the program by overwriting the original program end instruction (see Appending Instructions below).

10.8 DISPLAY Mode

STOP NOP m ENT / STOP NOP ENT

These commands switch the system to DISPLAY mode, allowing you to display the program instructions stored in memory one after another via the LEDs, by pressing the + and - keys to step forwards and backwards through the program. The MARK, POS, TIM, OUT, IF and NOP instructions are displayed directly with the LEDs of the corresponding keys; the DEL . instruction is displayed with the MARK and NOP LEDs together. The ERR LED lights up in the event of errors and at the end of the program (program end instruction). Pressing the CLR key exits the DISPLAY mode.

10.9 Insert Instruction

INS ENT

The DISPLAY mode makes it simple to move to the position where you want to insert one or more additional instructions. To insert instructions, proceed as follows: First, switch to DISPLAY mode and locate the instruction **BEFORE** which you want to insert the new instruction(s). Press CLR to exit the DISPLAY mode, and then press INS ENT once for each instruction you wish to insert. This inserts NOPs, which you can later overwrite with the appropriate instructions.

10.10 Delete Instruction

DEL ENT

In the same way as for inserting instructions, first switch to DISPLAY mode and locate the position at you wish to begin deleting, then press CLR to exit DISPLAY. Now press DEL ENT once for each instruction to be deleted. Each time you press DEL ENT, the current instruction is cleared from memory and all the following instructions are moved up by one line, allowing you to go on deleting towards the end of the program.

10.11 Appending Instructions

Existing programs can be lengthened very easily by overwriting the program end instruction. First, switch to DISPLAY mode and locate the end of the program. The ERR LED will light up. Now press CLR to switch to INPUT mode, and simply enter the additional instructions normally. Don't forget to enter the INS NOP ENT sequence again when you are finished!

10.12 Overwriting Instructions

This function is used to overwrite previously-entered NOPs, existing instructions or NOPs entered directly with the INS ENT sequence. Switch to DISPLAY mode and locate the instruction you wish to overwrite, then press the CLR key and enter the new instructions directly.

10.13 Program Run / Break

You can choose between running the program normally or in single-step mode.

RUN m ENT

Start normal program execution at label m

STOP CLR

Break program execution at any point

RUN ENT

Resume program execution at point where it was interrupted

The instruction sequence RUN NOP m ENT selects single-step mode, allowing you to test the program by running it one instruction at a time. You can then step through the program with the + key. Each time you press +, the next instruction in the memory is executed, moving the robot axes and lighting up the corresponding LED on the Teachbox.

11 Hardware Installation, Repackaging

11.1 Hardware requirements

To work with *TR 5* you need at least.

- The *TR 5*
- The power supply unit with two power leads (red/black)
- The Teachbox
- The RS-232 shorting connector

11.2 Installation

- Connect the *TR 5* to the power supply unit with the two power leads.
- Plug the Teachbox's 25-pin male connector into the I/O port and the 9-pin shorting connector into the RS-232 port.
- Turn the power switch to ON, and press the RESET key on the *TR 5*.

Now you can enter a program or move the *TR 5*'s arm directly with the appropriate instructions.

The *TR 5*'s program memory is non-volatile, and is powered by a long-life lithium battery; your programs will remain stored in the robot's memory for around 10 years, even when the mains power is switched off.

IMPORTANT!

Never try to change the position of the robot's arm manually, not even when no power is connected, as this can damage the *TR 5*'s mechanisms.

11.3 Packing the robot

To pack the robot for transportation, use the Teachbox or *TBPS* to position the arm as follows:

Axis 1	128
Axis 2	164
Axis 3	140
Axis 4	182
Axis 5	14
Gripper	128

Table 11-1 Packing position of *TR 5*

12 Digital Inputs and Outputs, EMERGENCY STOP Function

12.1 Digital Inputs (TTL)

INPUT	PIN
DI1:	8
DI2:	20
DI3:	7
DI4:	19
DI5:	6
DI6:	18
DI7:	5
DI8:	17

Table 12-1

All the digital inputs are active LOW

12.2 Outputs (TTL)

OUTPUT	PIN
DO1:	25
DO2:	12
DO3:	24
DO4:	11
DO5:	23
DO6:	10
DO7:	22
DO8:	9

Table 12-2

All the digital outputs are active LOW

12.3 EMERGENCY STOP for the TR 5

INPUT	PIN
DI10:	4

Table 12-3

The EMERGENCY STOP input is active LOW

The emergency off function is realised with an NOC across pins 4 and 1 of the robot's I/O port. Activation of the function switches off the TR 5's motors. After eliminating the problem, you must execute a RESET on the TR 5 unit, then you can restart the control program.

12.4 Power Supply

<i>OUTPUT</i>	<i>PIN</i>
GND	3 and 16
+ 5V	2 and 13 (I ≤ 50 mA)

Table 12-4

The power supply unit is only suitable for the connection of opto couplers. Any external driver stages must be provided with their own power supply. Please note that the ground lines should NOT be connected when opto couplers are being used.

13 Exercises with the Teachbox

Connect the *TR 5* and switch it on as described in Section 11, Hardware Installation, Repackaging. Before you begin the exercises, we suggest that you first move the robot's arm to the following home position:

AXIS	POS
Axis 1	128
Axis 2	100
Axis 3	160
Axis 4	160
Axis 5	128
Gripper	160

Table 13-1

To position the first axis, enter the following key sequence:

```
POS 1 NOP 128 ENT
```

When you press ENT, axis 1 of the *TR 5* will immediately move into position 128, which is the central position. Repeat the procedure with the appropriate values for the remaining five axes.

CAUTION!

Please be extremely careful when you position axes 2, 3 and 4 directly, as these axes can move the gripper below the plane of the surface on which the *TR 5* is mounted, which can result in gripper damage. Positions 180-255 with axis 2 are particularly dangerous. If the gripper ever does collide the tabletop when you are positioning the arm directly, press the STOP key on the Teachbox immediately and hold it down until you have pressed the RESET key on the *TR 5*. This initialises the last arm position as the current position. The *TR 5* will then stop in this position, and you can change the position by entering further teach instructions. Please note that a reset clears all program instructions entered beforehand from the robot's memory, and they must then be re-entered from the start.

If the red ERR lamp on the Teachbox lights up while you are entering instructions, first clear the error condition by pressing the CLR key, then re-enter the last instruction sequence.

All the following exercises are entered with label 0 as the starting address, and can be started by entering RUN 0 ENT. If you wish, however, you can also enter other starting addresses, so long as they are within the permitted value range (0 <= starting address <= 118).

13.1 Program Header and Program End

STOP 0 ENT

This sequence generates the program header, clearing the contents of the TR 5's memory and initialising it for programming. It should only be entered once, at the beginning of a new programming session. DON'T use this command when you want to edit programs already in memory or add new programs to the those in memory, as it will completely erase all the instructions and programs already stored in the robot's memory!

MARK 0 ENT

Set label 0 as the starting address, marking the beginning of your program.

CLR

Switch to POSITION mode

This is followed by the body of the program, consisting of positioning sequences, loops and input and output instructions.

INS NOP ENT

This sequence must always be entered at the end of every taught-in program. When you are editing existing programs, it should never be entered if you make changes in the body of the program only, but you **MUST** enter it if you lengthen the program by overwriting the original program end instruction and adding more instructions.

14 Practice Programs

14.1.1 PROGRAM 1 Single Axis Positioning

STOP 0 ENT	Clear memory, generate program header
MARK 0 ENT	Enter start address
CLR	Switch to INPUT mode
POS ENT	Store home arm position as starting position

(All the practice programs begin with these four lines)

1	Select axis 1 for TEACH positioning
+	Depress key to turn right in TEACH mode
CLR	Switch to INPUT mode
POS ENT	Store new position as position 2
1	Select axis 1 for TEACH positioning
-	Depress key to turn left in TEACH mode
CLR	Switch to INPUT mode
POS ENT	Store new position as position 3
POS 1 NOP 150 ENT	Move axis 1 to position 150
CLR	Switch to INPUT mode
POS ENT	Store new position as position 4
POS 1 NOP 50 ENT	Move axis 1 to position 50
CLR	Switch to INPUT mode
POS ENT	Store new position as position 5
INS NOP ENT	Program end

This program uses two different methods for entering single axis movements: Direct teaching, using the + and - keys, and direct entry of the target position with the POS (axis a) . (position n) ENT instruction sequence. Both of these methods can be used for all the robot's axes.

14.1.2 PROGRAM 2: Consecutive Single Axis Positioning for all 6 Axes

STOP 0 ENT	Clear memory, generate program header
MARK 0 ENT	Enter start address
CLR	Switch to INPUT mode
POS ENT	Store home arm position as starting position

(Standard beginning of the program)

POS 1 NOP 150 ENT	Move axis 1 to position 150
CLR	Switch to INPUT mode
POS ENT	Store new position as position 2
2	Select axis 2 for TEACH positioning
-	Depress key to move up in TEACH mode
CLR	Switch to INPUT mode
POS ENT	Store new position as position 3
POS 3 NOP 50 ENT	Move axis 3 to position 50
CLR	Switch to INPUT mode
POS ENT	Store new position as position 4
4	Select axis 4 for TEACH positioning
-	Depress key to move up in TEACH mode
CLR	Switch to INPUT mode
POS ENT	Store new position as position 5
POS 5 NOP 10 ENT	Move axis 5 to position 10
CLR	Switch to INPUT mode
POS ENT	Store new position as position 6
6	Select axis 6 for TEACH positioning
-	Press key to open gripper in TEACH mode
CLR	Switch to INPUT mode
POS ENT	Store new position as position 7
INS NOP ENT	Program end

In the above example, all of the robot's axes are moved one after another in a consecutive sequence. When you start the program, all the axes will first be moved simultaneously to bring them back into the original home position, and then the programmed movement sequence will be repeated.

14.1.3 PROGRAM 3: Simultaneous Multiple Axis Positioning

Note!

No matter how many axes you move, the new robot arm position is not stored in the memory until you enter the POS ENT sequence.

STOP 0 ENT	Clear memory, generate program header
MARK 0 ENT	Enter start address
CLR	Switch to INPUT mode
POS ENT	Store home arm position as starting position

(Standard beginning of the program)

POS 1 NOP 150 ENT	Move axis 1 to position 150
2	Select axis 2 for TEACH positioning
-	Press key to move up in teach mode
CLR	Switch to INPUT mode
POS 3 NOP 50 ENT	Move axis 3 to position 50
4	Select axis 4 for TEACH positioning
-	Press key to move up in TEACH mode
5	Select axis 5 for TEACH positioning
-	Press key to turn right in TEACH mode
6	Select axis 6 for TEACH positioning
-	Press key to open gripper in TEACH mode
CLR	Switch to INPUT mode
POS ENT	Store new position as target position
INS NOP ENT	Program end<R>

Now that you have familiarised yourself with beginning and terminating programs and positioning single and multiple axes, you can start adding other instructions to your programs. Please consult Sections 6 (Summary of all Instructions and Commands) and 7 (Reference to the Individual Instructions) for full details.

14.1.4 PROGRAM 4: Sample Program

STOP 0 ENT	Clear memory, generate program header
MARK 0 ENT	Enter start address
CLR	Switch to INPUT mode
POS ENT	Store home arm position as starting position
TIM 40 ENT	Set delay, 4 seconds
POS 1 NOP 150 ENT	move axis 1 to position 150
CLR	Switch to INPUT mode
POS ENT	Store new position as position 2
IF 1 ENT	Wait until input 1 is set (LOW = GND)
2	Select axis 2 for TEACH positioning
-	Press key to move up in TEACH mode
CLR	Switch to INPUT mode
POS 3 NOP 50 ENT	Move axis 3 to position 50
4	Select axis 4 for TEACH positioning
-	Press key to move up in TEACH mode
CLR	Switch to INPUT mode
POS ENT	Store new position as position 3
TIM 20 ENT	Set delay, 2 seconds

(The following sequence switches another machine or device on, leaves it on for 10 seconds then switches it off again.)

OUT 1 + ENT	Set output 1 to LOW = GND
TIM 100 ENT	Set delay, 10 seconds
OUT 1 - ENT	Reset output 1 to HIGH = +5V
GOTO 0 NOP 5 ENT	Loop to label 0 5 times
INS NOP ENT	Program end

14.2 Tips and Suggestions

You should also set a delay of 1/2 second before gripping or releasing something with the gripper. This gives the gripper time to come to rest, ensuring that the object will be gripped and released smoothly and cleanly.

We strongly advise that you first test your programs in single-step mode after teaching them in. Start the program with RUN . 0 ENT; you can then step through the program one instruction at a time with the + key. All the instructions (MARK, POS, TIM etc.) will be displayed on the LEDs of the Teachbox as they are executed, starting with the first MARK instruction and ending with the program end instruction (the latter is indicated by the ERR LED). At the same time, the robot will execute the programmed movements.

15 Include Files

The Include files contain procedures which help you to write your control programs for the *TR 5* in BASIC, Pascal and C, controlling the robot via the COM1 or COM2 interface of your computer.

15.1 Supported Compilers and Interpreters.

These procedures can be integrated into any program generated with the following compilers or interpreters:

- **QBASIC from Microsoft**

The compiler (or interpreter)-specific procedures must be edited for use with other Basic compiler or interpreter!

- **Turbo Pascal 6.0 from Borland**

The compiler-specific procedures must be edited for use with other Pascal compilers!

- **Microsoft C 6.0**

The compiler-specific procedures must be edited for use with other C compilers!

Under the current copyright legislation, it is not possible to supply the software compilers and interpreters together with the *TR 5* software.

15.1.1 Summary of the TR 5 Include Files

15.1.1.1 BASIC-Files

The following files are required for creating BASIC programs using the BASIC Include procedures. All the files are stored in straight ASCII format.

include.bas	TR 5 subprograms
head.bas	Program header for TR 5 programs
bas_demo.bas	Demo program in BASIC

15.1.1.2 PASCAL-Files

The following files are required for writing Pascal programs with the Pascal Include procedures.

inc_pas.def	All constant definitions
inc_pas.par	Global variables
include.pas	TR 5 procedures
head.pas	Program header for TR 5 programs

A demo program is also included on the diskette, both in compiled form and as source code.

pas_demo.pas	Pascal demo program (source code)
pas_demo.com	Executable Pascal demo program

15.1.1.3 C-Files

The following files are required for writing C programs using the C Include procedures.

inc_c.def	Constant definitions
inc_c.par	Global variables
include.c	TR 5 procedures
head.c	Program header for TR 5 programs.

A demo program is also included on the diskette, both in compiled form and as source code.

c_demo.c	C demo program (source code)
c_demo.exe	Executable C demo program

15.1.2 How to use the header files

The distribution disk contains the following header files:

- HEAD.BAS
- HEAD.PAS
- HEAD.C

These files contain standard program headers, making the creation of the headers for your TR 5 programs a fast and simple matter. If you are using the Turbo Pascal editor or WordStar (for example), all you need to do is open a new file and read in the appropriate header file from the disk. The BASIC header file HEAD.BAS is a special case: If you edit this file as an ASCII file, you can proceed as described above. In addition, however, you can also merge the file into an existing BASIC program using the command MERGE "HEAD BAS". In the latter case it is important to remember that any lines in your program with the same line numbers as the lines in the HEAD.BAS file will be overwritten!

Purpose of the Header Files:

The header files contain include directives for inserting all the necessary Include files in your program (applies for HEAD.PAS and HEAD.C only).

These files are as follows:

HEAD . BAS	none
HEAD . PAS	INC_PAS . DEF INC_PAS . PAR INCLUDE . PAS
HEAD . C	STDIO . H DOS . H INC_C . DEF INC_C . PAR INCLUDE . C

All the header files also specify the serial port device to be used (COM1 or COM2), and contain code for initialising the PC and TR 5 interfaces.

Important!

Before running control programs written in C or Turbo Pascal, you must ALWAYS run either RS_ROB1.COM or RS_ROB2.COM. This is not necessary for programs written in QBASIC.

15.2 The BASIC Include Subprograms

15.2.1 SUBPROGRAM 1: Set Specified Robot Axis Position

Function	Moves the axis AXIS to position POSITION.	The TR 5 then returns its status in STATUS.
Subprogram Call	GOSUB 51000	
Parameter 1	AXIS:integer (1..6)	Axis number
Parameter 2	POSITION: integer (0..255)	Target position of the selected axis
Return Parameter	STATUS: integer (0,1)	Can be used for evaluation of TR 5 status: 1 = Positioning executed 0 = Positioning could not be executed

15.2.2 SUBPROGRAM 2: Set all Robot Axis Positions

Function	Moves all axes to the positions defined in POSARRAY	TR 5 then returns its status
Subprogram Call	GOSUB 52000	
Parameter	POSARRAY: Field (1..6) integer	Target positions (0..255) of the axes (1..6)
Return Parameter	STATUS: integer (0,1)	Can be used for evaluation of TR 5 status: 1 = Positioning executed 0 = Positioning could not be executed

15.2.3 SUBPROGRAM 3: Read Specified Robot Axis Position

Function	Polls axis AXIS and returns its position	
Subprogram Call	GOSUB 53000	
Parameter	AXIS: integer (1..6)	Axis number
Return Parameter	VAR: integer (0..255)	Contains the returned position of the specified axis

15.2.4 SUBPROGRAM 4: Read all Robot Axis Positions

Function	Polls all the axes and returns their positions	
Subprogram Call	GOSUB 54000	
Return Parameter	POSARRAY: Field (1..6) integer	Contains the returned axis positions

15.2.5 SUBPROGRAM 5: Read Input Port Value

Function	Reads the value from the digital input port;	Resolution: 8 bits
Subprogram Call	GOSUB 55000	
Parameter	PORT = "4"	
Return Parameter	VAR: integer	Contains the byte (8 bits) returned from the digital input

15.2.6 SUBPROGRAM 6: Write Byte to Output Port

Function	Writes a byte (8 bits) to the digital output port	The TR 5 does not return any parameters
Subprogram Call	GOSUB 56000	
Parameter	PORTVAL: integer (0..255)	Output to the digital output port

15.2.7 SUBPROGRAM 7: Output Port Byte AND Parameter Byte

Function	Performs a logical AND on the byte at the output port and the byte in the parameter, and writes the result to the output port.	The TR 5 does not return any parameters
Subprogram Call	GOSUB 57000	
Parameter	PORTVAL: integer (0..255)	

15.2.8 SUBPROGRAM 8: Output Port Byte OR Parameter Byte

Function	Performs a logical OR on the byte at the output port and the byte in the parameter, and writes the result to the output port.	The TR 5 does not return any parameters
Subprogram Call	GOSUB 58000	
Parameter	PORTVAL: integer (0..255)	

15.2.9 SUBPROGRAM 9: Output Port Byte XOR Parameter Byte

Function	Performs a logical XOR on the byte at the output port and the byte in the parameter, and writes the result to the output port.	The TR 5 does not return any parameters
Subprogram Call	GOSUB 59000	
Parameter	PORTVAL: integer (0..255)	

15.2.10 SUBPROGRAM 10: Set Output Port Bit

Function	Sets bit BIT at the digital output port.	TR 5 does not return any parameters
Subprogram Call	GOSUB 60000	
Parameter	BIT: integer (0..7)	Bit to set

15.2.11 SUBPROGRAM 11: Clear Output Port Bit

Function	Clears bit BIT at the digital output port	TR 5 does not return any parameters
Subprogram Call	GOSUB 61000	
Parameter	BIT: integer (0..7)	Bit to clear

15.2.12 SUBPROGRAM 12: Initialise PC Interface

Function	Initialises the PC interface as follows:	
	Interface:	COM1 or COM2
	Baud Rate:	9600, 4800, 2400, 1200
	Parity:	NONE
	Data Bits:	8
	Stop Bits:	1
Subprogram Call	GOSUB 47000	

15.2.13 SUBPROGRAM 13: Initialise TR 5 Interface

Function	Initialises the TR 5 interface.	The TR 5 ascertains the baud rate automatically.
Subprogram Call	GOSUB 50000	

15.2.14 SUBPROGRAMS 14 and 15:

Subprogram Call	GOSUB 48000 , GOSUB 49000	
-----------------	------------------------------	--

These subprograms are required by the other TR 5 subprograms. Please don't use or change them yourself.

15.3 The Pascal and C Include Procedures**15.3.1 PROCEDURE 1: Set Specified Robot Axis Position**

Function	Moves axis AXIS to position POS; the instruction is ignored if the axis or position parameters are invalid.	TR 5 returns its status in STATUS
Procedure Name	SET_POS (AXIS , POS , STATUS)	
Parameter 1	AXIS: integer (1..6)	Axis number
Parameter 2	POS: integer (0..255)	Target position of selected axis
Return Parameter	STATUS: integer (0,1)	Can be used for evaluation of TR 5 status: 1 = Positioning executed 0 = Positioning could not be executed

15.3.2 PROCEDURE 2: Set all Robot Axis Positions

Function	Moves all axes to the positions defined in POSARRAY; the instruction is ignored if the parameters include invalid position values.	TR 5 then returns its status
Procedure Name	SET_POS_ALL(POS_ARRAY , STATUS)	
Parameter	POSARRAY: array[1..6] of integer	Target positions (0..255) of all axes
Return Parameter	STATUS: integer (0,1)	Can be used for evaluation of TR 5 status: 1 = Positioning executed 0 = Positioning could not be executed

15.3.3 PROCEDURE 3: Read Specified Robot Axis Position

Function	Polls axis AXIS and returns its position; the instruction is ignored if the axis parameter contains an invalid value.	
Procedure Name	READ_POS(AXIS , POS)	
Parameter	AXIS: integer (1..6)	Axis number
Return Parameter	POS: integer (0..255)	Contains the returned position of the specified axis

15.3.4 PROCEDURE 4: Read all Robot Axis Positions

Function	Polls all the axes and returns their positions	
Procedure Name	READ_POS_ALL(POS_ARRAY)	
Return Parameter	POS_ARRAY: ARRAY [1..6] of integer	Contains the returned axis positions

15.3.5 PROCEDURE 5: Read Input Port Value

Function	Reads the value from the digital input port;	Resolution: 8 bits
Procedure Name	READ_PORT(4 , VALUE)	
Parameter	CONSTANT "DIGITAL"=4	
Return Parameter	VALUE: integer	Contains the byte (8 bits) returned from the digital input

15.3.6 PROCEDURE 6: Write Byte to Output Port

Function	Writes a byte (8 bits) to the digital output port	The <i>TR 5</i> does not return any parameters
Procedure Name	SET_PORT (VALUE)	
Parameter	VALUE: integer (0..255)	Output to the digital output port

15.3.7 PROCEDURE 7: Output Port Byte AND Parameter Byte

Function	Performs a logical AND on the byte at the output port and the byte in the parameter, and writes the result to the output port.	The <i>TR 5</i> does not return any parameters
Procedure Name	SET_AND_PORT (VALUE)	
Parameter	VALUE: integer (0..255)	

15.3.8 PROCEDURE 8: Output Port Byte OR Parameter Byte

Function	Performs a logical OR on the byte at the output port and the byte in the parameter, and writes the result to the output port.	The <i>TR 5</i> does not return any parameters
Procedure Name	SET_OR_PORT (VALUE)	
Parameter	VALUE: integer (0..255)	

15.3.9 PROCEDURE 9: Output Port Byte XOR Parameter Byte

Function	Performs a logical XOR on the byte at the output port and the byte in the parameter, and writes the result to the output port.	The <i>TR 5</i> does not return any parameters
Procedure Name	SET_XOR_PORT (VALUE)	
Parameter	VALUE: integer (0..255)	

15.3.10 PROCEDURE 10: Set Output Port Bit

Function	Sets bit BIT at the digital output port; the instruction is ignored if an invalid bit value is specified in the parameter.	<i>TR 5</i> does not return any parameters
Procedure Name	SET_BIT (BIT)	
Parameter	BIT: integer (0..7)	Bit to set

15.3.11 PROCEDURE 11: Clear Output Port Bit

Function	Clears bit BIT at the digital output port; the instruction is ignored if an invalid bit value is specified in the parameter.	TR 5 does not return any parameters
Procedure Name	SET_NOT_BIT (BIT)	
Parameter	BIT: integer (0..7)	Bit to clear

15.3.12 PROCEDURE 12: Initialise PC Interface

Function	Initialises the PC interface	
Procedure Name	INIT_COM (COM , BAUD RATE , PARITY , BITS , STOP)	
Parameter 1	COM: integer (COM1,COM2)	Communications port specification: COM1 or COM2
Parameter 2	BAUDRATE: integer (300..9600)	Sets communication speed: 300 - 9600 Baud
Parameter 3	PARITY: char (E,O,N)	Parity: EVEN, ODD, NONE
Parameter 4	BITS: integer (7,8)	Data bits: 7, 8
Parameter 5	STOP: integer (1,2)	Stop bits: 1, 2

15.3.13 PROCEDURE 13: Initialise TR 5 Interface

Function	Initialises the TR 5 interface.	TR 5 ascertains the baud rate automatically.
Procedure Name	SENDE_SPACE	

16 The vacuum gripper

16.1 The vacuum-gripper consists of the following components:

- Control box with integrated suction pump
- Gripper device with rubber tool
- Power supply (12V)
- Suction tube
- Self-adhesive cable fixing

16.2 Mounting and connecting the items

1. Move the gripper of the robot in position 200:

POS 6 NOP 200

2. Put up the gripper device on the robots gripper and fix it with the two screws.
3. Connect the rubber tool with the suction terminal of the control box by way of the pneumatic tube. It is recommended to fix the tube with the self-adhesive cable fixings on the upper side of the robot arm. Take sure that the mechanics have enough freedom of movement. The tube must not be squeezed.
4. Connect the 25-pin SUB-D plug of the control box either with the 25-pin plug of the Teachbox (Teachbox-mode) or with the 25-pin terminal of the robots I/O port (software-mode).
5. Connect the power cable of the control box with the power supply.

CAUTION!

Don't confuse the polarity!

red cable= +

black cable= -

16.3 Running the system

After switching on the power supply of the robot as well as the power supply of the vacuum gripper the system is ready for use.

The suction pump can be activated by setting the digital output 1 (Teachbox or TBPS software or by setting the digital output 0 (PC I/O card).

For an optimised function of the gripper it is recommended that the user includes a delay of about 1 second before the robot lift up the workpiece

16.4 The control box

CLAMP	PORT	Remark
1	12V	from power supply
2	GND	from power supply
3	Output 0 [1]	+12V if digital output 0 = HIGH or open
4	Output 0 [1]	+12V if digital output 0 = LOW
5	Output 0 [1]	GND
6	Output 1 [2]	+12V if digital output 0 = HIGH or open
7	Output 1 [2]	+12V if digital output 0 = LOW
8	Output 1 [2]	GND
9	Output +5V	pin 13 of the 25 pin I/O port of the robot
10	Output 0 [1]	pin 25 of the 25 pin I/O port of the robot
11	Output 1 [2]	pin 12 of the 25 pin I/O port of the robot

The values in the brackets are valid for TBPS and the Teachbox.

Output 1 [2] mentioned in the table above is free for a 12V user-device. It can be controlled by the digital output 1 [2] of the robot.

ATTENTION! The capability of the power supply must not be overload.

16.5 Programming the vacuum gripper with Teachbox and TBPS

To control the gripper output 1 of the robot is used.

Activate the vacuum gripper:

OUT 1 + ENT

Deactivate the vacuum gripper:

OUT 1 - ENT

SAMPLE:

STOP 0 ENT	Clear memory, generate program header
MARK 0 ENT	Enter start address
CLR	Switch to INPUT mode
POS ENT	Store home arm position as starting position
OUT 1 + ENT	Set digital output 1 of the robot (activate the vacuum pump)
TIM 50 ENT	Delay of 5 sec. (the vacuum gripper is 5 sec. activated)

OUT 1 - ENT	Reset digital output 1 of the robot (deactivate the vacuum pump)
TIM 20 ENT	Delay of 2 sec. (the vacuum gripper is 2 sec. deactivated)
GOTO 0 NOP 5 ENT	Jump 5 times to mark 0
INS NOP END	End program

16.6 Programming the vacuum gripper with PSI

To use the vacuum gripper with the PSI software, the gripper should be defined with the TOOL DEFINE instruction (see *PSI* manual for details)

TOOL NAME	free
RESOLUTION	0
TOOL CO-ORDINATES	free

SAMPLE:

TOOL----	VACUUM	GRIPPER	1		
COORD	x	+49.00	y	+0.00	z +0.00
	a	+0.00	b	+0.00	c +0.00
REMARK	activate vacuum gripper				
PORT->	L	0			
REMARK	deactivate vacuum gripper				
PORT->	H	0			

17 I/O-Card SMARTLAB**17.1 General information**

The digital inputs and outputs of the robot controllers for the robot types TR5 and IR 50 p are TTL-compatible and not opto-decoupled. To allow peripheral equipment being not TTL-compatible (this applies especially to PLCs) to be connected to the controller and to simultaneously decouple these periphery components, a corresponding interface is required. This function is taken over / covered by the PC I/O card type SMARTLAB. This card is used as interface between the robot and the peripheral equipment to be connected to it.

The PC-I/O card can be used together with the programming system for industrial robots "PSI", the TeachBox programming system "TBPS" and the software library being part of the TBPS-package.

**To operate the I/O card, please
make sure that the robot is in
ONLINE mode !**

17.2 Standard scope of delivery:

- 1 I/O plug-in card for PC
- 2 connecting cables (PC internal) card ↔ SUB D
- 1 x 37-pin male connector (incl. connector shell)
- 1 x 37-pin female connector (incl. connector shell)

17.3 Technical data:

Inputs	The card offers 16 digital, opto-decoupled inputs being independent of each other (max. 45 VDC) at a 37-pin D-SUB-male connector
Outputs	The card features 16 potential-free, digital relay outputs being independent of each other (150 VDC/1A) at a 37-pin D-SUB female connector
Power supply	Power supply for the PC I/O card is ensured via the AT bus of the PC's power supply unit. To supply the peripheral equipment with power, an appropriate, separate power supply unit is required. This power supply unit is not part of the PC I/O-card's scope of delivery!

Table 17-1 Technical data of the I/O-card

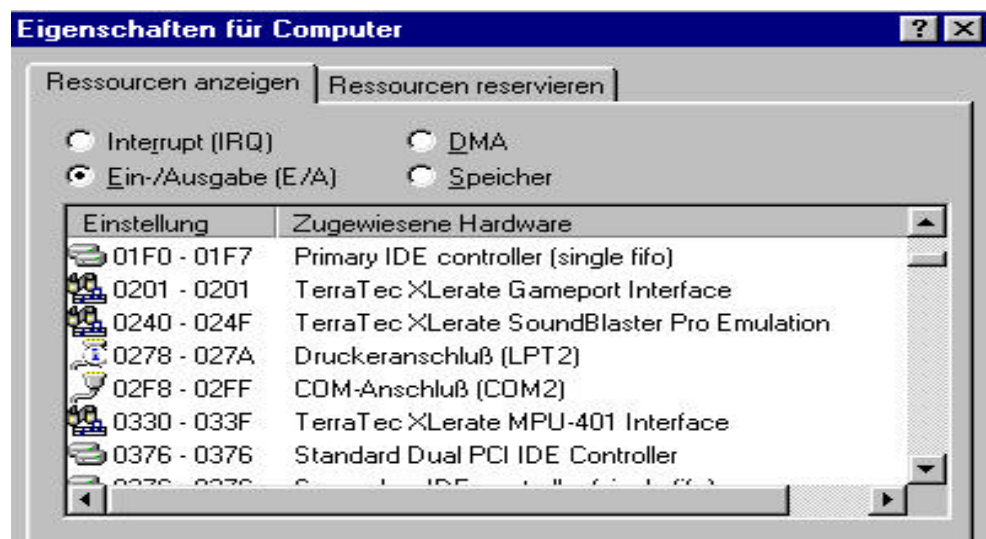
17.4 Commissioning

The I/O-card does not support a *Plug and Play mode*. Consequently, the necessary settings must be made „manually“ by the operator.

Before installing the card in your PC and in case you work with MS *WINDOWS* operating systems, you should first determine the free I/O addresses of your PC.

On the display you reach by selecting the menu options: **Start > Settings > System control > Systems > Device manager (Properties) | In- / output**

the I/O addresses already occupied on your PC are visualized.



When searching for free addresses, preference shall be given to searching in the area ranging between 0200h and 0300h. On the I/O-card, the address 0270h has been preset as default value. In case that this address on your PC is not yet occupied, you may continue with commissioning following the instruction given under *1.4 Installation*.

If the address 0270h is shown as being already occupied, the next possible, free address range should be selected.

Note: please see also the I/O-card manual, pages 1.21 to 1.24!

In the manual, you will find a graphical representation of the addresses that can be set / adjusted by means of DIP-switches. According to the selected address, please set the DIP-switches on the I/O card to the position indicated.

The jumpers to set the IRQ must be all open.

In case that address conflicts occur despite having followed the procedure described above that may be due to other cards plugged into your PC, please get immediately in touch with *P@P Picking Systems*.

17.5 Installation

- Switch off the PC and remove all connecting cables.
- Carefully open the PC-housing.
- Remove 3 slot cover sheets next to each other at the position of your PC housing where you later intend to mount the card and the SUB-D connector (female connector).
- Plug the stake connector J1 on the corresponding plug connector J1. Fix the stake connector such that the red core of the flat cable is connected to pin 1 of the plug connector. Repeat this procedure for stake connector J2 and plug connector J2.
- Now insert the I/O-card into the expansion port of your PC.
- Fix the two SUB-D connectors (female connector) and screw the card tight.
- Close the PC and plug in again all connecting cables required.
- You may now switch on your PC and check for its proper functioning.

17.6 Configuring PSI

To be able to operate the I/O-card with PSI, first a change has to be made in one of the PSI configuration files. Therefore, please proceed as described in the following:

- Open the file PSI_conf.cfg in the PSI directory by means of using a text editor (e.g. NOTEPAD or WORDPAD in *MS WINDOWS*).
- Now modify the line starting with \$,5 as follows: **\$,5,SMARTLAB, 624** (The number "624" is the **decimal** address of the I/O-card)

```
# PSI CONFIGURATION FILE
#
$,4,com2:19200,8,n,1
# parallel I/O type
# valid entrys:
# SMARTLAB, base port address in decimal
#     (parallel I/O is done by PC I/O CARD SMARTLAB)
#     (default value is 624)
$,5,SMARTLAB, 624
#$,6,LINEAR
#lin-axis is connected to the robot-controller ($,6,LINEAR)
#lin-axis is not connected to the robot-controller (#$,6,LINEAR)!
# !!! I M P O R T A N T !!!
#if you want to start PSI, only to move the robot without the lin-axis, then
#remark this line as follows:
#     #,6,LINEAR
```

- Now save the file you modified accordingly and start PSI.

The digital in- and outputs of the card can now be addressed by the following PSI commands:

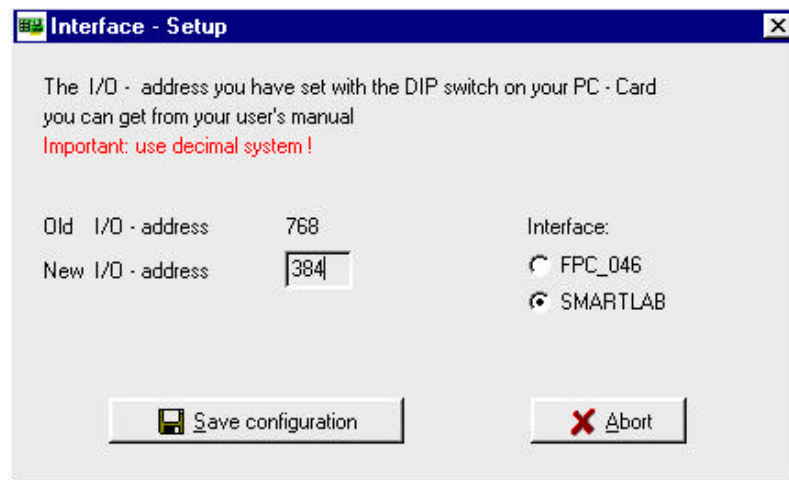
PORT	->	H	XX
PORT	->	L	XX
IF	PORT	XX=	HIGH
IF	PORT	XX=	LOW JUMP

(XX= PORT number)

17.7 Configuring TBPS for Windows

To be able to operate the I/O-card with TBPS for Windows, it is necessary to carry out the following steps:

- Set the port address of the I/O-card accordingly as described in the section before.
- Start the program **Interface-Setup** and enter the **decimal** I/O address accordingly.



17.8 Drawings

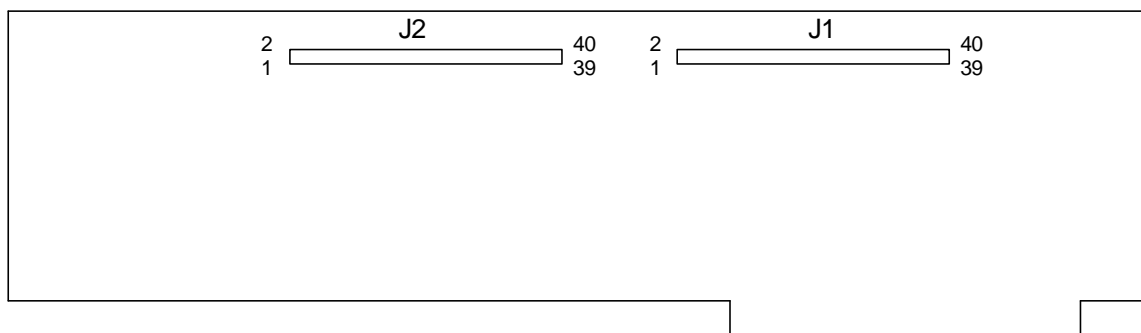


Figure 17-0 Layout and position of the plug contacts of the I/O-card

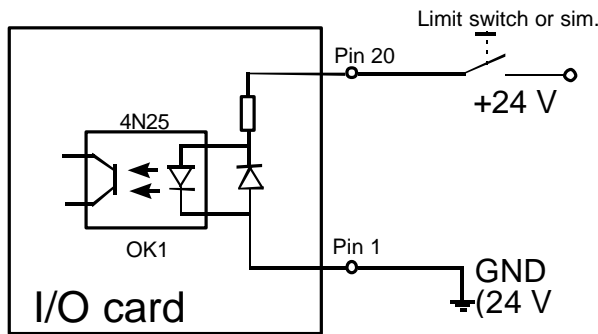


Figure 17-1

Example for an input wiring of the I/O-card

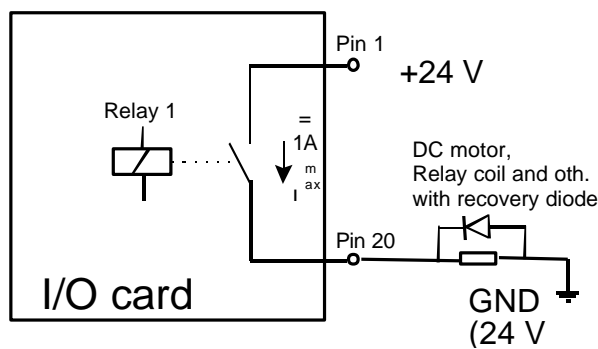


Figure 17-2

Example for an output wiring of the I/O-card

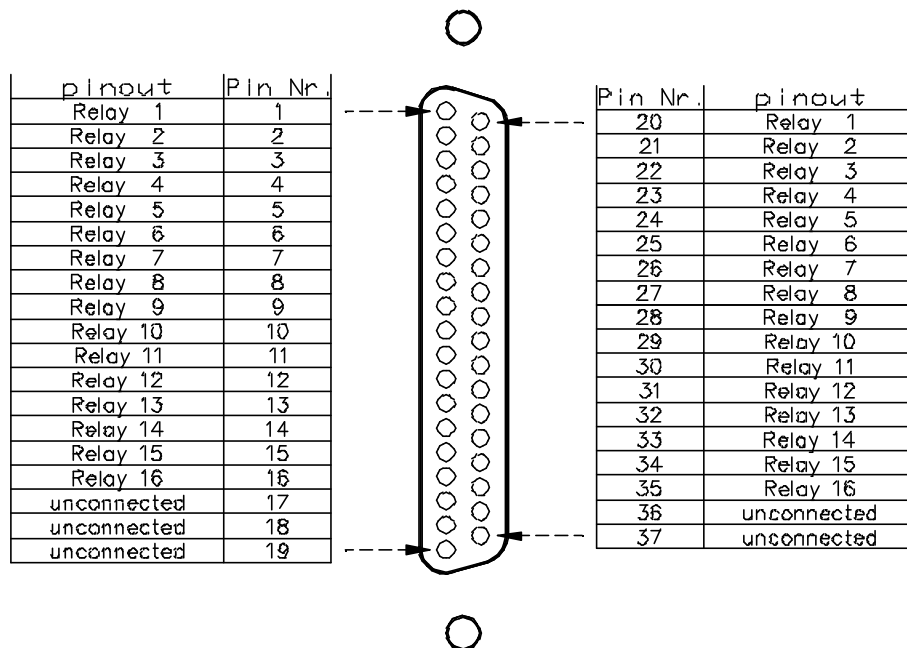


Figure 17-3 Pin assignment of the 37-pin SUB-D female connector at the sheet metal covering the slot

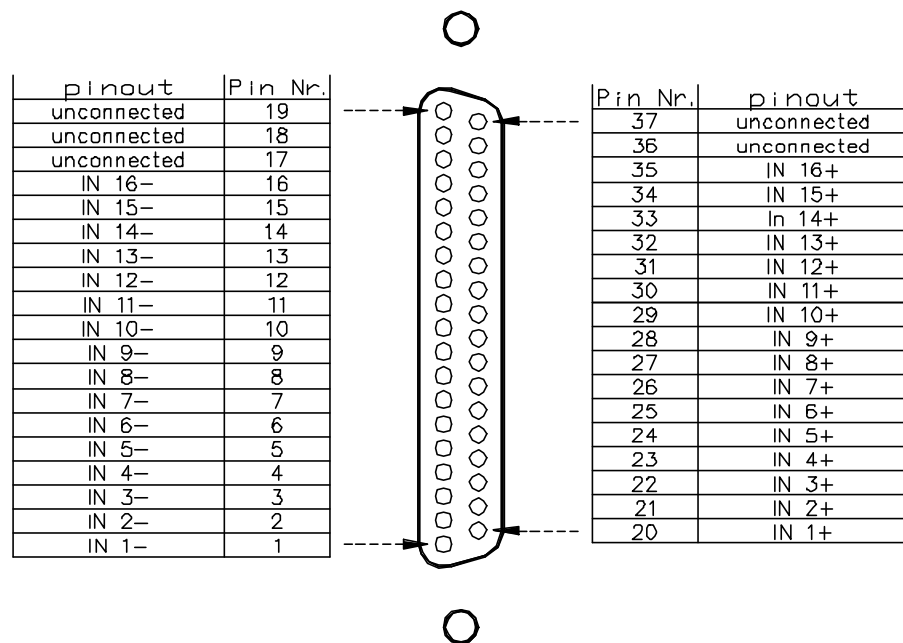


Figure 17-4 Pin assignment of the 37-pin SUB-D male connector at the metal sheet covering the slot

18 Limited Warranty Terms and Conditions

Warranted against defects in materials and workmanship for a period of 12 (twelve) months as of the date of purchase, delivery free works. Wearing parts such as motors and potentiometers are excluded from this Warranty coverage. The use of physical force, incorrect connection and any and all alterations or repair attempts on the part of the Customer will render this Warranty null and void.

We reserve the right to make technical alterations without notice!

18.1 Shipping address for service and repair

P@P Picking Systems GmbH
Schafhofstraße 10

D-90411 Nürnberg (GERMANY)

Tel.: ++49 - 911 – 955 119-0

Fax.: ++49 - 911 – 955 119-11

eMail: info@pp-systeme.com

Homepage: www.pp-systeme.com

Space for additional notes and addresses:

19 Index of Figures

Figure 3-1 <i>TR 5</i>	6
Figure 3-2 The <i>TR 5</i> controller	7
Figure 3-3 The standard gripper equipped with the vacuum option	7
Figure 3-4 The hand held Teachbox.....	8
Figure 3-5 The power supply for <i>TR 5</i>	9
Figure 3-6 The RS 232 interface cables	10
Figure 3-7 The I/O Testbox.....	10
Figure 3-8 The PC I/O-Card.....	11
Figure 5-1 Dimensions of <i>TR 5</i>	16
Figure 6-1 Working range of <i>TR 5</i>	17
Figure 7-1 Teachbox.....	18
Figure 17-0 Layout and position of the plug contacts of the I/O-card.....	52
Figure 17-1 Example for an input wiring of the I/O-card	53
Figure 17-2 Example for an output wiring of the I/O card.....	53
Figure 17-3 Pin assignment of the 37-pin SUB-D female connector at the sheet metal covering the slot.....	53
Figure 17-4 Pin assignment of the 37-pin SUB-D male connector at the metal sheet covering the slot.....	54

20 Index of Tables

Table 4-1	12
Table 4-2	13
Table 4-3	13
Table 4-4	14
Table 4-5	15
Table 9-1	20
Table 9-2	20
Table 11-1 Packing position of TR 5.....	25
Table 12-1.....	26
Table 12-2.....	26
Table 12-3.....	26
Table 12-4.....	27
Table 13-1.....	28
Table 17-1 Technical data of the I/O-card.....	49